

---

# EECS 427

## Lecture 10: Multipliers and Recap

Readings: 11.4, WH 10.9

# Reminders

---

- Quiz 1 on Wednesday 10/14
  - 70-minute quiz: 1:40-2:50 pm
  - One-page cheat sheet, non-programmable calculator
  - Coverage until the end of the multiplier section
  - Zhengya's office hours: today 3-4 pm, Tu 3:30-5:30 pm, no office hour on Wed. Other time available by appointment.
- CAD4 is due Wednesday 10/14
  - You can submit it by Thursday 10/15 at noon

# Last Time

---

- Multiplication consists of AND functions to generate the partial products and lots of additions
  - Carry and sum delays of adder cells can be equally critical
- Carry-save multiplier shortens critical path
  - Increases one more adder stage
- Modified Booth recoding reduces the # of partial products to be added, improves speed
  - Also suitable for 2s complement multiplication

# Overview

---

- Review modified Booth recoding
- Sign extension and pre-computation
- Tree multipliers
  
- Quiz 1 review

## Recoding Table

---

From MSB to LSB, look at multiplier bits two at a time and in conjunction with the MSB of adjacent less-significant pair

$Y_{2i+1}$	$Y_{2i}$	$Y_{2i-1}$	$PP_i$	X: multiplicand Y: multiplier
0	0	0	0	
0	0	1	X	
0	1	0	X	start of string of 1s
0	1	1	2X	
1	0	0	-2X	
1	0	1	-X	why -x?
1	1	0	-X	end of string of 1s
1	1	1	0	

Overlap with next pair of bits

EECS 427 F09

Lecture 10

5

## Example

---

$10110010 \quad (-78)$   
 recode  $\rightarrow 10011101 \quad (-99)$

---

$111111110110010$   
 $00000001001110$   
 $11110110010$   
 $001001110$

---

~~1~~0001111000101010  $(7722)$

EECS 427 F09

Lecture 10

6

# In-class example

---

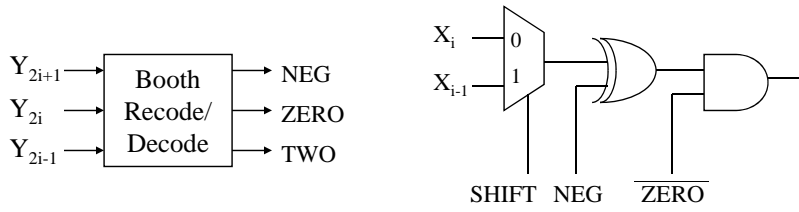
$$\begin{array}{r}
 \phantom{X} \quad 010111 \\
 X \quad 011110 \\
 \hline
 \end{array}$$

# Booth Decoding and Partial Product Generation

---

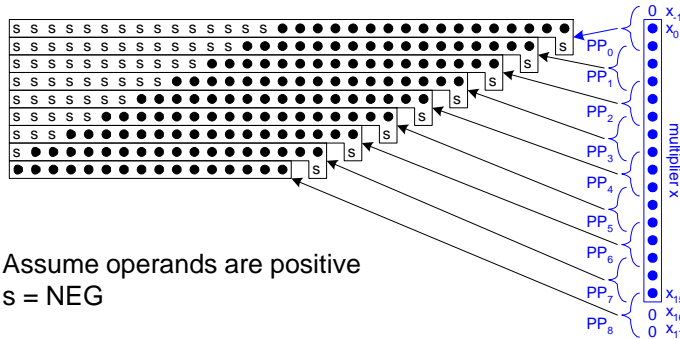
X: multiplicand  
Y: multiplier

Operation	NEG	ZERO	SHIFT
0	0	1	0
X	0	0	0
-X	1	0	0
2X	0	0	1
-2X	1	0	1



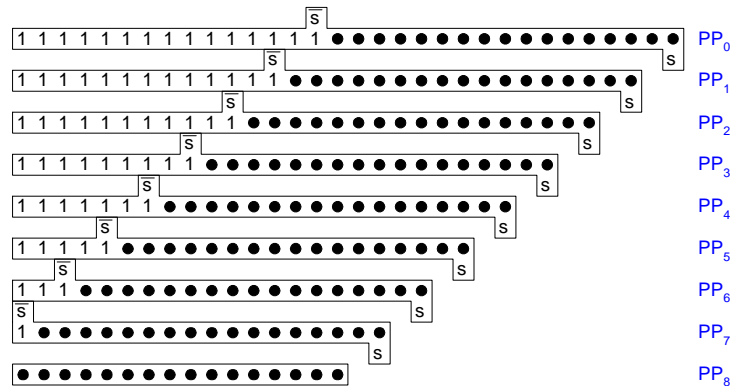
# Sign Extension

- Partial products can be negative
  - Require sign extension, which is cumbersome



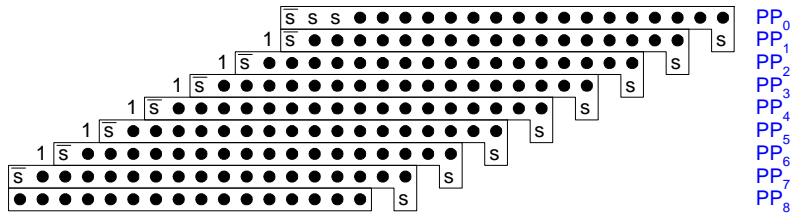
# Simplified Sign Ext.

- Sign bits are either all 0's or all 1's
  - Note that all 0's is all 1's + 1 in proper column



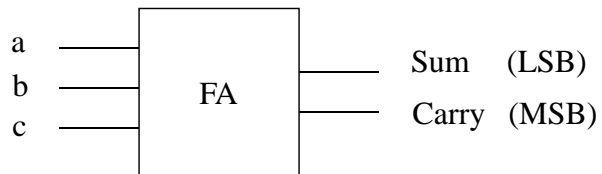
## Even Simpler Sign Ext.

- No need to add all the 1's in hardware
  - Precompute the answer!



## 3:2 Compressor

Encode the sum of 3 bits using 2 bits



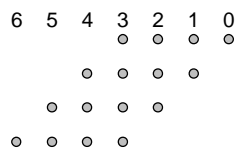
# 3:2 Compressor

In constant time, convert the sum of 3 n-bit numbers a, b, d into a sum of 2 n-bit numbers s and c



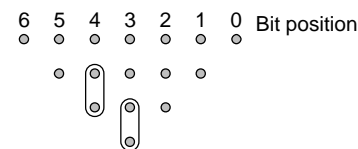
# Wallace Tree Multiplier

Partial products



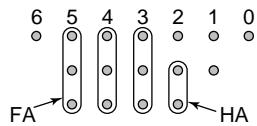
(a)

First stage



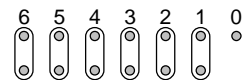
(b)

Second stage



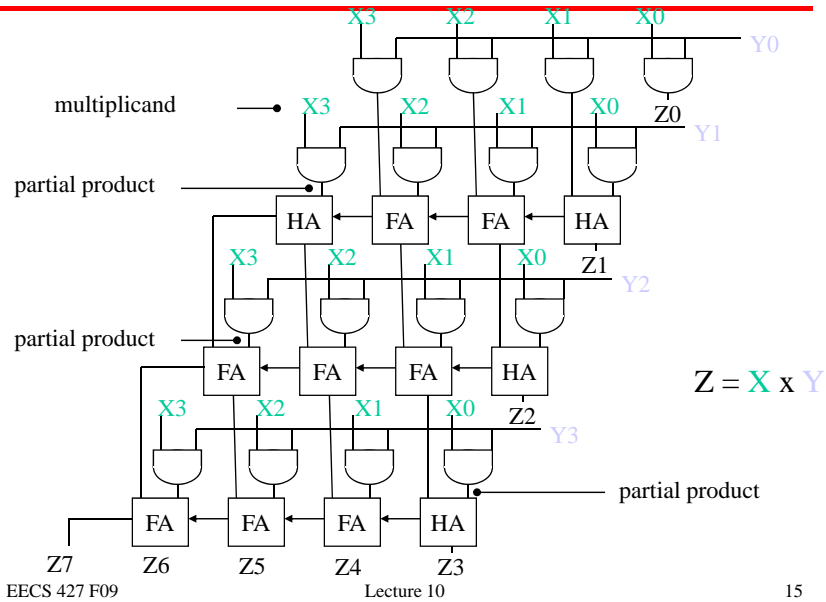
(c)

Final adder

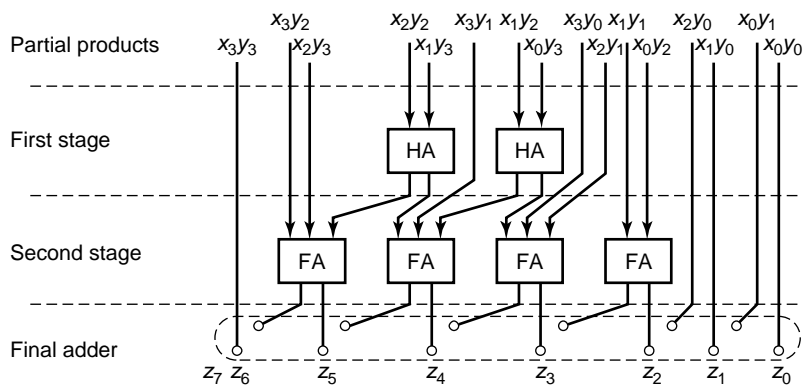


(d)

# Array Multiplier



# Wallace Tree Multiplier



- In  $\log_{3/2}N$  levels, tree of (3:2) compressors can transform sum of  $N$  numbers into sum of 2 numbers

# Summary

---

- Modified Booth recoding reduces the # of partial products to be added, improves speed
  - Also suitable for 2s complement multiplication
- Tree structures reduce the # of adders needed and delay increases only logarithmically with # of bits
  - Inefficient layout structure
- Could pipeline the datapath to increase performance

# Quiz 1 Review

---

- ----- Background -----
- Lecture 1: CMOS process and fabrication
- Lecture 2: Design rule and layout
  
- ----- Design and sizing of logic gates -----
- Lecture 3: CMOS steady state (VTC), transient (R, C, delay), complementary CMOS (fan-in, fanout)
- Lecture 4 and 5: Logical effort (intrinsic, fanout delay, balancing stage effort, optimal # of stages, systematic sizing recipe, limitations)
- Lecture 6 and 7: Logic styles (static: pseudo-NMOS, DCVSL, PTL and level restoration, CPL, Transmission gate, dynamic: keeper, domino, footless, dual-rail, etc.)
  
- ----- Arithmetic building blocks -----
- Lecture 8: Adder logic (P, G, D, S, Cout, duality, mirror adder, transmission gate, Manchester), topology (ripple, carry bypass, carry select), lookahead (P, G, radix-2/radix-4, full/sparse trees)
- Lecture 9 and 10: Shifters (barrel, logarithmic), multipliers (array, carry-save, modified booth recoding, sign extension, tree multiplier)